

Visual Domain Specific Languages for Actuarial Models: An Industrial Experience Report

Workshop on Domain Specific Languages for Financial Systems

ACM/IEEE 16th International Conference on Model Driven Engineering Languages and Systems (MODELS 2013), Miami, FL

Aon Benfield Securities, Inc.
Annuity Solutions Group (ASG)

October 1, 2013

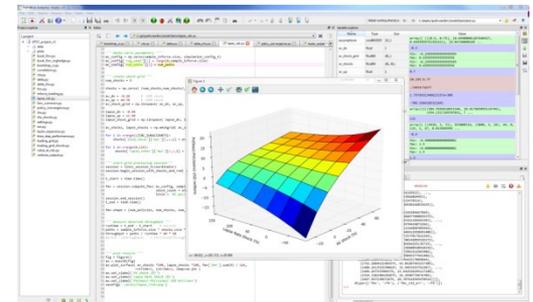
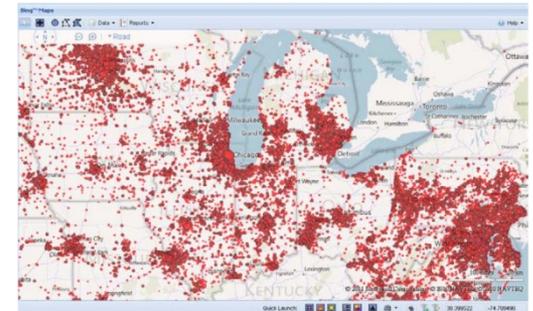


This document is the confidential property of Aon Benfield Securities, Inc. (“Aon”), has been prepared by Aon for informational purposes only and is intended only for the designated recipient. As a condition to reviewing this document, the recipient agrees that without the prior written consent of Aon, which may be withheld for any reason, the recipient will not copy the document or any of its contents, and will not disclose or disseminate the document or any of its contents to (i) any third party, or (ii) any person within recipient’s organization who does not have a need to know in connection with the express business purpose for which the document is being provided to recipient. If the recipient is legally compelled to disclose this document or any of its contents, it will promptly give notice to Aon and will reasonably cooperate with Aon in any attempts by Aon to obtain a protective order or otherwise limit disclosure. Upon request by Aon, the recipient will promptly return or destroy the document and any copies it has made with Aon’s consent (as described above), provided that recipient may maintain, in strict confidentiality, such copy or copies as required by law or regulation. Aon makes no representation of any kind as to the suitability of the products or services described in this document for any entity in any jurisdiction. The recipient is advised to undertake its own review of the legal, regulatory, tax, accounting and actuarial implications of the products and services described in this document, as Aon does not provide legal, regulatory, tax, accounting or actuarial opinions. This document should not be considered an offer to sell or a solicitation of any agreement to purchase any security. All securities advice, products or services are offered solely through Aon Benfield Securities, Inc. or an appropriately licensed affiliate.

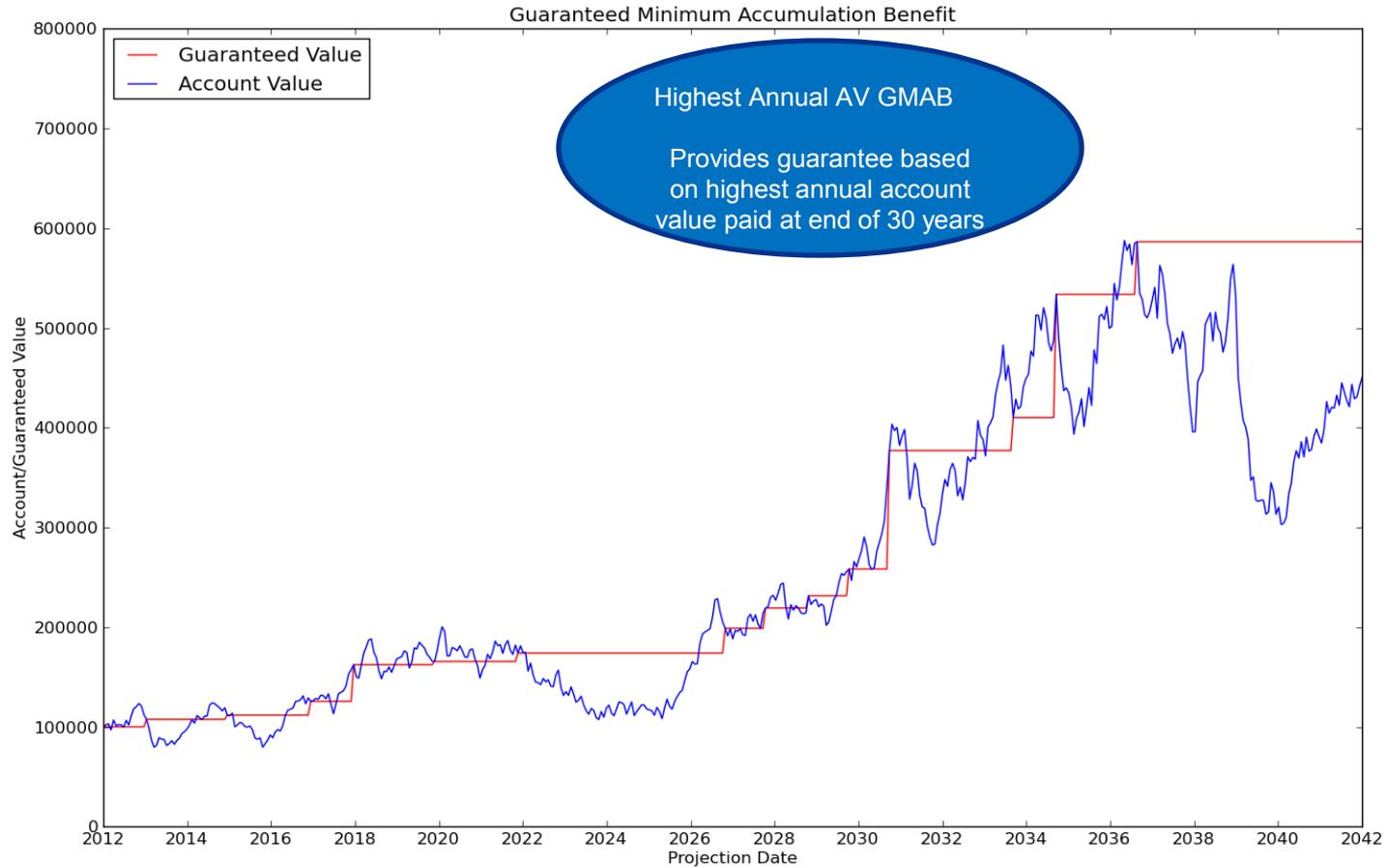


Aon Benfield

- **Aon Benfield**, a division of Aon plc (NYSE: AON), is the world's leading reinsurance intermediary and full-service capital advisor. We empower our clients to better understand, manage and transfer risk through innovative solutions and personalized access to all forms of global reinsurance capital across treaty, facultative and capital markets. As a trusted advocate, we deliver local reach to the world's markets, and an unparalleled investment in innovative analytics. With more than 80 offices in 50 countries, our worldwide client base has access to the broadest portfolio of integrated capital solutions and services.
- **Aon Benfield Analytics**
 - Aon Benfield Analytics offers clients industry-leading catastrophe management, actuarial, rating agency advisory and risk and capital strategy expertise.
 - Sample risk analytics products
 - PathWise
 - ReMetrica
 - ImpactOnDemand
 - CatScore



Industry Overview – Variable Annuities



Industry Computational Challenges

- **Business end-users focus**
 - ♦ Users are Quantitative Analysts, Actuaries, Traders, Risk Managers, etc
 - ♦ The right tools must focus on the end-user requirements
- **Business logic and systems code must be continually adapted to changes**
 - ♦ Changing models, financial products, market conditions, and regulatory requirements
 - ♦ Changing technologies (Multi-Core, Cell Broadband Engine, GPUs, etc)
- **High Computational Throughput is required**
 - ♦ Large-scale real-time Monte Carlo simulations (Support Hedging Programs)
 - ♦ Nested simulations (Hedging Back Testing, Capital, Valuation)
 - ♦ High end-user productivity (not waiting for huge runs to complete)
- **Mission Critical Operations**
 - ♦ The intended use of such systems is mission critical
 - ♦ System failures or bugs can be catastrophic for business users
 - ♦ Automation and auditability are very important issues

Industry Computational Challenges

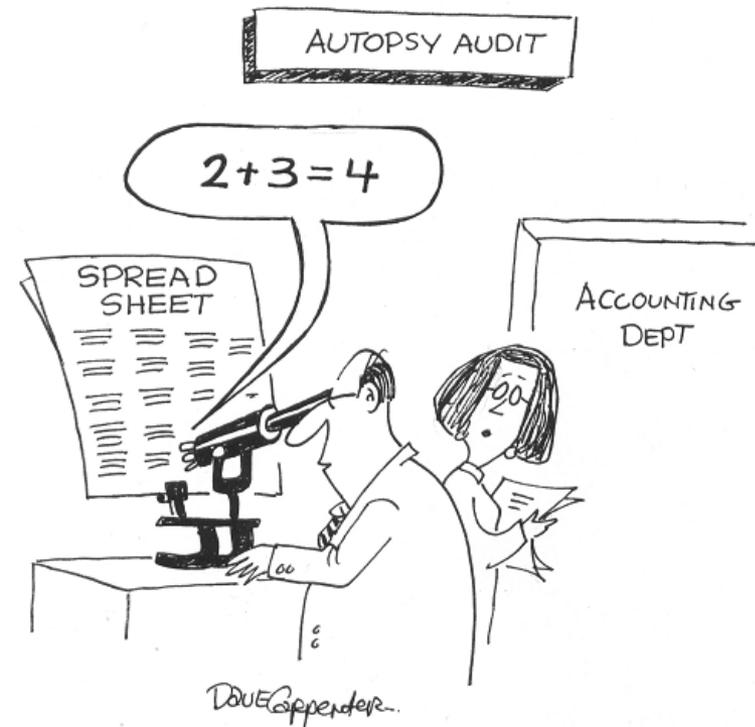
- **Business logic and systems code must be continually adapted to changes**

- Change is constant

- Financial modeling innovation
- Financial products innovation
- Evolving market conditions
- Changing regulatory requirements
- Technological innovation

- Traditional approaches

- Enterprise IT systems slow to adapt
- Shadow IT systems fill the gaps – patchwork of end-user developed, manually operated spreadsheets (potentially thousands of interlinked spreadsheets)
- Slow, costly, error-prone



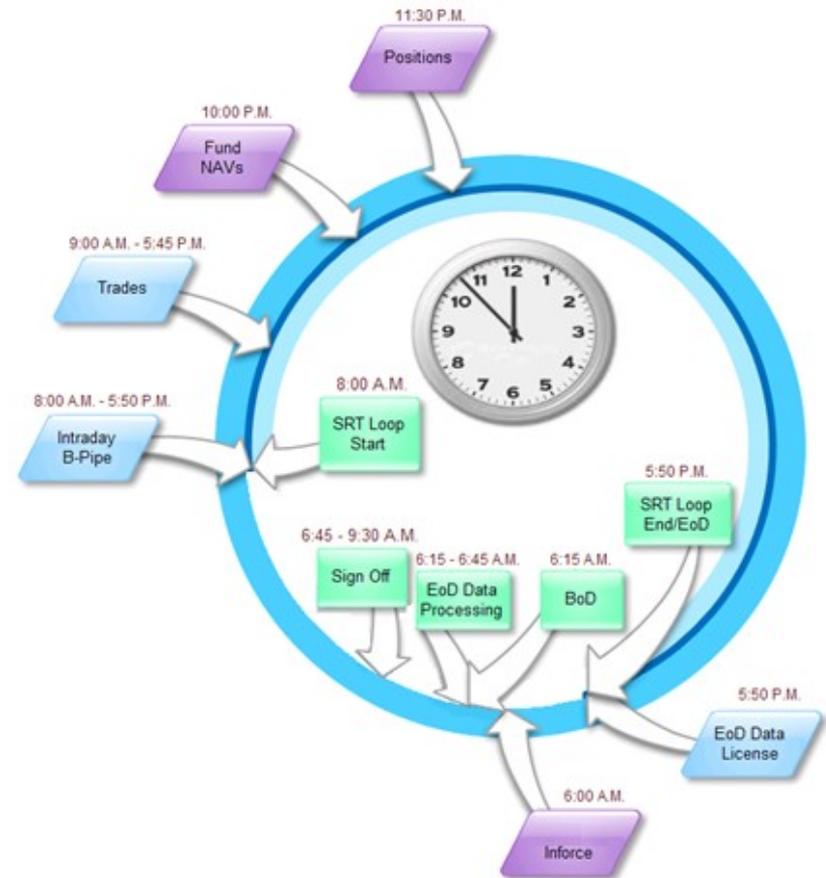
"There it is! I've isolated the origin of the firm's demise."

Industry Computational Challenges

■ Mission Critical Operations

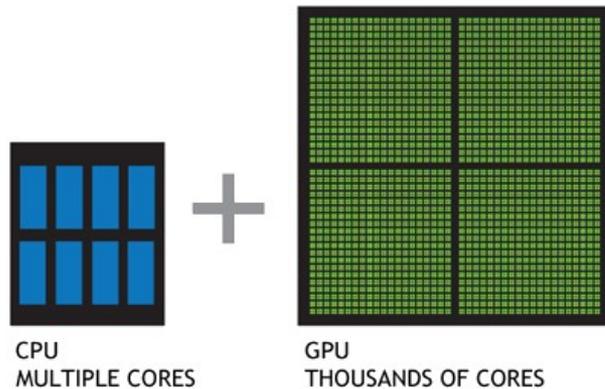
– Requirements

- High performance, integrated real-time analytics
- Complex business data-flow management
- Job scheduling
- Fault tolerance / failover
- Operational workflows
- Reporting presentation layers
- Audit trails
- Monitoring and Error Reporting



GPU Computing

- GPU (Graphics Processing Units) are specialized processors that can be used to speed-up parallel computing problems, such as Monte Carlo simulation



– Implications for Variable Annuities Modeling:

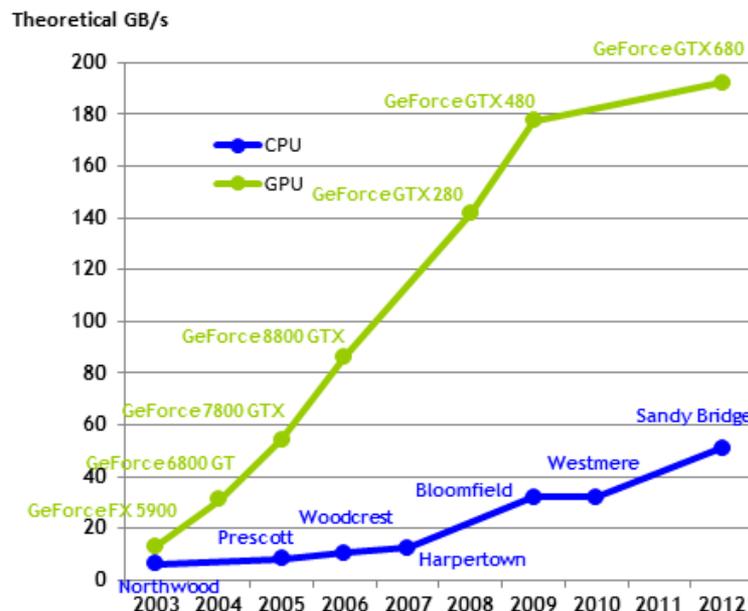
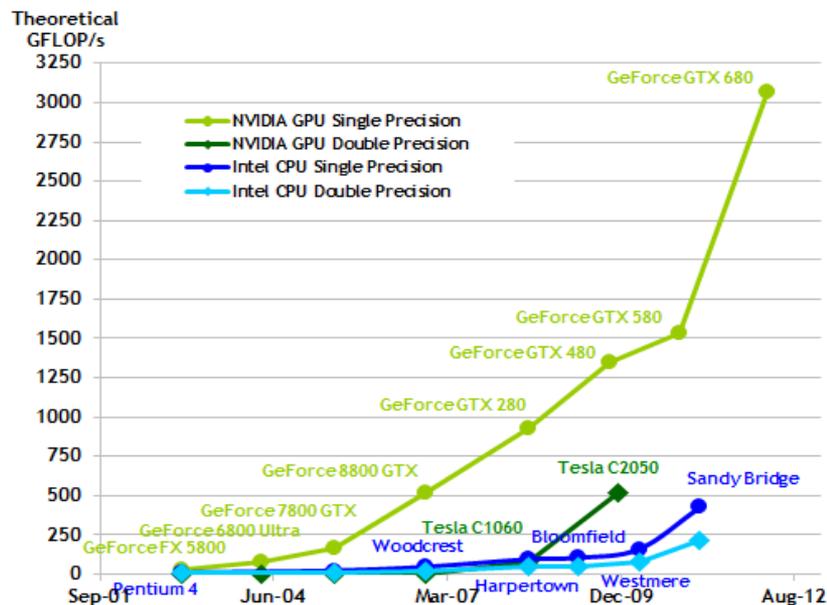
- **50-500x** speed improvements for stochastic models, when compared to equivalent CPU-based software

Above: Tesla K10 GPU module, containing 3,072 cores

Source: <http://www.nvidia.ca/object/what-is-gpu-computing.html>

GPU Computing

- Performance gap between CPUs and GPUs continues to increase rapidly



– Implications for Variable Annuities Modeling:

- Cost of GPU-based grids is increasingly lower than cost of equivalent CPU-based grids
- Complex optimizations for GPUs are increasingly important for VA modeling software (not simply a matter of farming out small sections of legacy code to GPUs)

GPU Computing

- **General Purpose Computing performance on GPUs continues to increase rapidly**

	Tesla M2050	Tesla K10	Improvement
Release Date	May-10	Dec-12	
Cores	448 cores	3072 cores	686%
Memory (GDDR5)	3GB	8GB	267%
Memory Bandwidth	148GB/s	320GB/s	216%
Single Precision Peak Performance	1.04 TFLOPS	4.58 TFLOPS	440%
Power Consumption	225W	235W	

- Tesla M2050 and Tesla K10 have similar hardware and power consumption costs
- Our VA modeling benchmarks show a 200-300% increase in efficiency (scenarios per second, per GPU or per dollar) when comparing Tesla M2050 GPUs in K10 GPUs

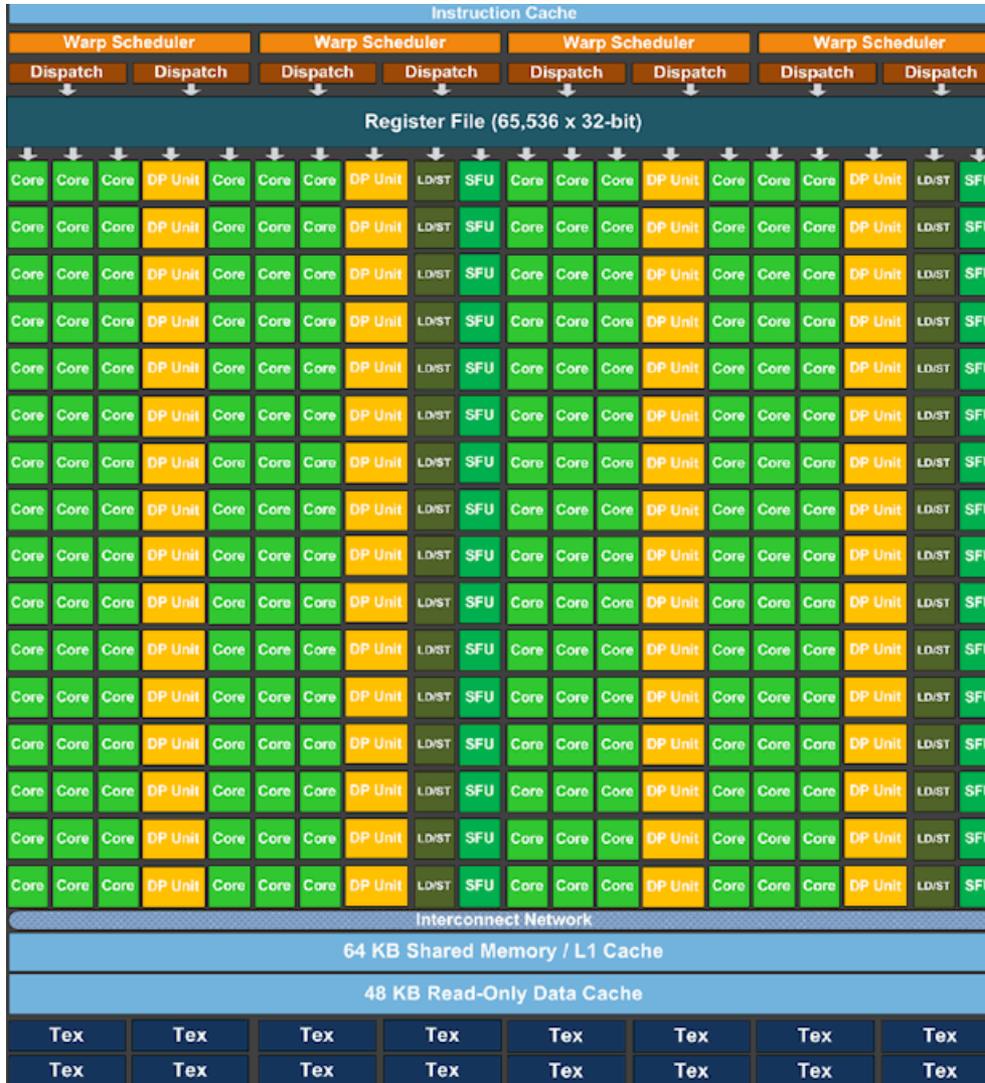
GPU Computing

NVIDIA Kepler GK110 processor



GPU Computing

GK110 processor SMX



GPU Computing

NVIDIA CUDA programming model

```
//Vector size in elements
const int N = 1048576;
//Vector size in bytes
const int dataSize = N * sizeof(float);

//CPU memory allocation
float *h_A = (float *)malloc(dataSize);
float *h_B = (float *)malloc(dataSize);
float *h_C = (float *)malloc(dataSize);

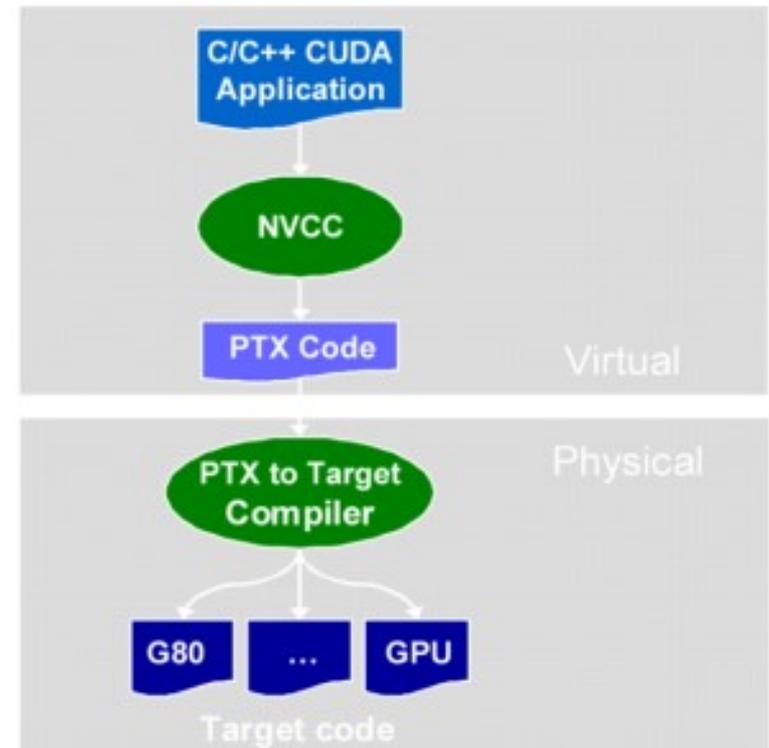
//GPU memory allocation
float *d_A, *d_B, *d_C;
cudaMalloc((void **)&d_A, dataSize);
cudaMalloc((void **)&d_B, dataSize);
cudaMalloc((void **)&d_C, dataSize);

//Initialize h_A[], h_B[]...

//Copy input data to GPU for processing
cudaMemcpy(d_A, h_A, dataSize, cudaMemcpyHostToDevice) );
cudaMemcpy(d_B, h_B, dataSize, cudaMemcpyHostToDevice) );

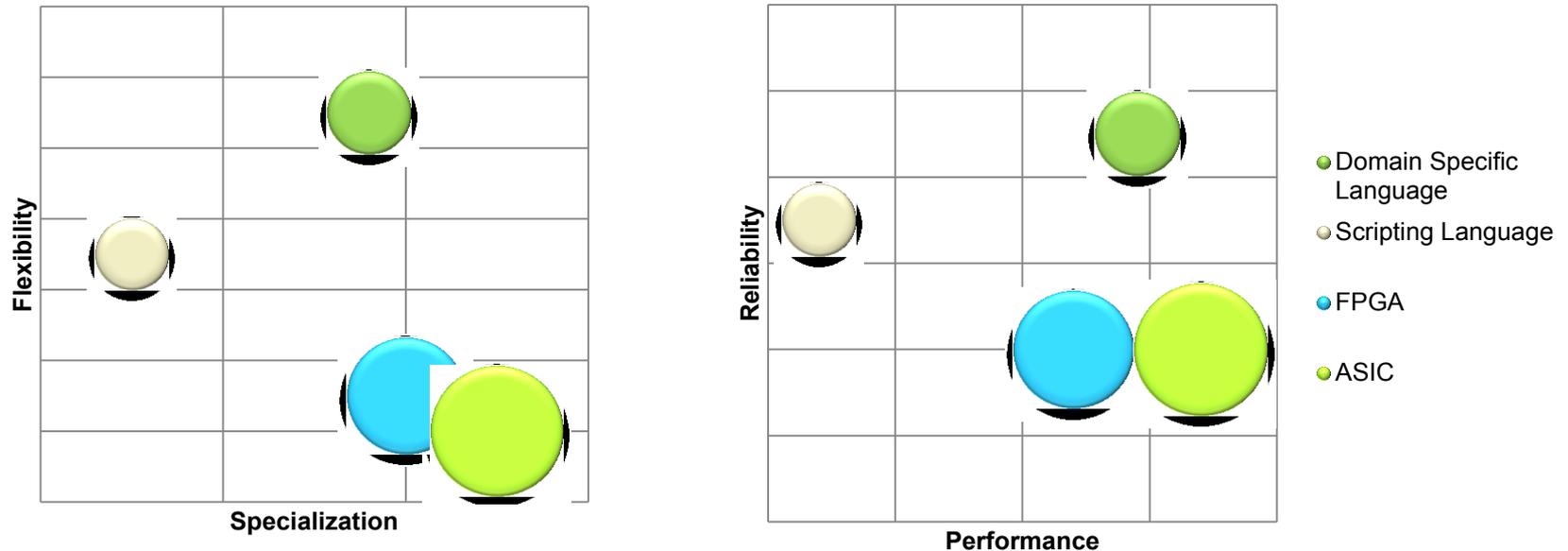
//Run the core of N / 256 units, 256 streams each
//Assuming that N is multiple of 256
vectorAdd<<<N / 256, 256>>>(d_C, d_A, d_B);

//Read GPU results
cudaMemcpy(h_C, d_C, dataSize, cudaMemcpyDeviceToHost) );
```



Domain Specific Languages

Example HPC Solution Trade-Offs

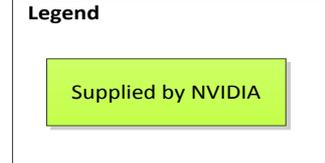
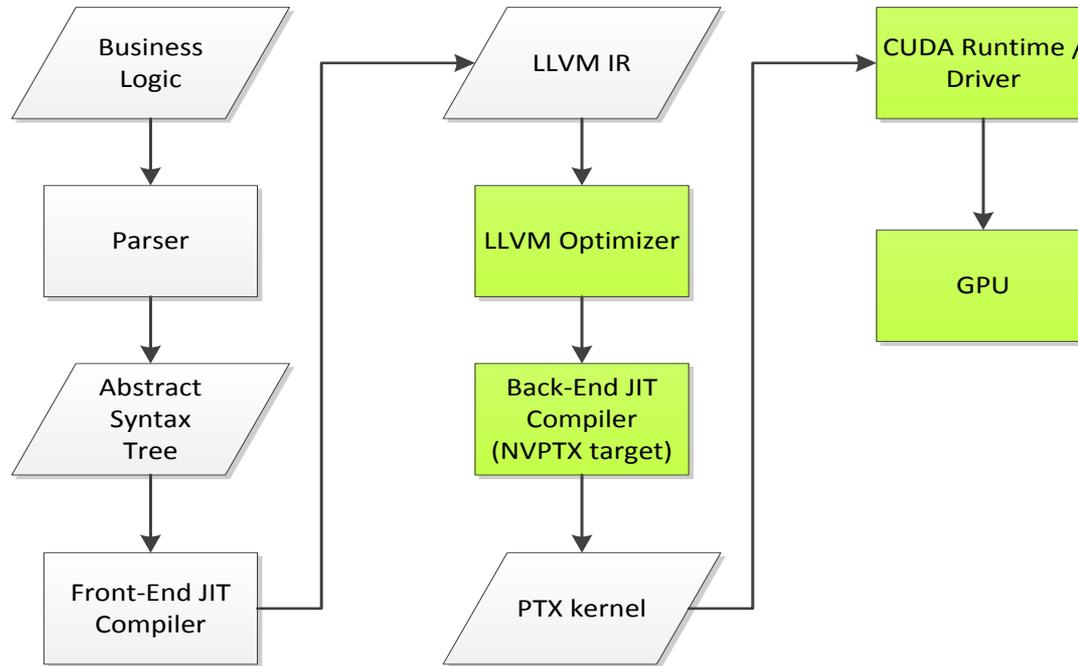


Size of bubble indicates cost (in terms of time and money) of solution

- **Flexibility** – ability to rapidly make changes
- **Specialization** – code specialized to specific hardware
- **Performance** – run-time performance of the solution
- **Reliability** – probable number of bugs in a large system

Domain Specific Languages

GPU DSL compiler architecture



Domain Specific Languages

GPU DSL compiler architecture

DSL

```
foo(x0, x1, x2)
{
    return x0 + x1 * (x2 + 1.0)
}
```

LLVM IR

```
; ModuleID = 'module1'
target triple = "nvptx64"

define double @foo(double %x0, double %x1, double %x2)
{
entry:
    %x23 = alloca double
    %x12 = alloca double
    %x01 = alloca double
    store double %x0, double* %x01
    store double %x1, double* %x12
    store double %x2, double* %x23
    %x04 = load double* %x01
    %x15 = load double* %x12
    %x26 = load double* %x23
    %faddtmp = fadd double %x26, 1.000000e+00
    %fmultmp = fmul double %x15, %faddtmp
    %faddtmp7 = fadd double %x04, %fmultmp
    ret double %faddtmp7
}
```

Domain Specific Languages

GPU DSL compiler architecture

PTX

```
//  
// Generated by LLVM NVPTX Back-End  
//  
  
.version 3.1  
.target sm_20, texmode_independent  
.address_size 64  
  
    // .globl      foo  
.entry foo(  
    .param .f64 foo_param_0,  
    .param .f64 foo_param_1,  
    .param .f64 foo_param_2  
)  
{  
  
    .local .align 8 .b8      __local_depot0[24];  
    .reg .b64      %SP;  
    .reg .b64      %SPL;  
    .reg .pred %p<396>;  
    .reg .s16 %rc<396>;  
  
...  

```

```
...  
  
    .reg .s16 %rs<396>;  
    .reg .s32 %r<396>;  
    .reg .s64 %rl<396>;  
    .reg .f32 %f<396>;  
    .reg .f64 %fl<396>;  
  
mov.u64      %SPL, __local_depot0;  
cvta.local.u64 %SP, %SPL;  
ld.param.f64 %f10, [foo_param_0];  
st.f64 [%SP+16], %f10;  
ld.param.f64 %f10, [foo_param_1];  
st.f64 [%SP+8], %f10;  
ld.param.f64 %f10, [foo_param_2];  
st.f64 [%SP+0], %f10;  
ld.f64 %f11, [%SP+16];  
ld.f64 %f12, [%SP+8];  
add.f64      %f10, %f10, 0d3FF0000000000000;  
fma.rn.f64   %f10, %f12, %f10, %f11;  
st.param.f64 [func_retval0+0], %f10;  
ret;  
  
}
```

PathWise Platform



PathWise Industry Recognition

INSURANCE & TECHNOLOGY

Sign Up for Our Free Subscription Now

Architecture / Infrastructure | Policy Administration & Management | Claims | Management Strategies | Security / Risk Management

Home
News
Blog / Opinion
Executive / Carrier Profiles
Case Studies

Standard Life's Traditional Prudence Drives Adoption of Aon Benfield's State-of-the-Art Annuity Risk Management Solution

- Insurance and Technology Magazine

“We had ready access to risk information on a regular basis before PathWise, but now the information is refreshed frequently and we’re able to make more timely decisions,” says Ettles.

“**Many calculations that we would have done in hours or days are now done every few minutes** — our information is up-to-date on a real-time basis and we’re not taking decisions on information that is stale.” Martin Ettles is a senior actuary, finance and risk management, Standard Life

PathWise Industry Recognition

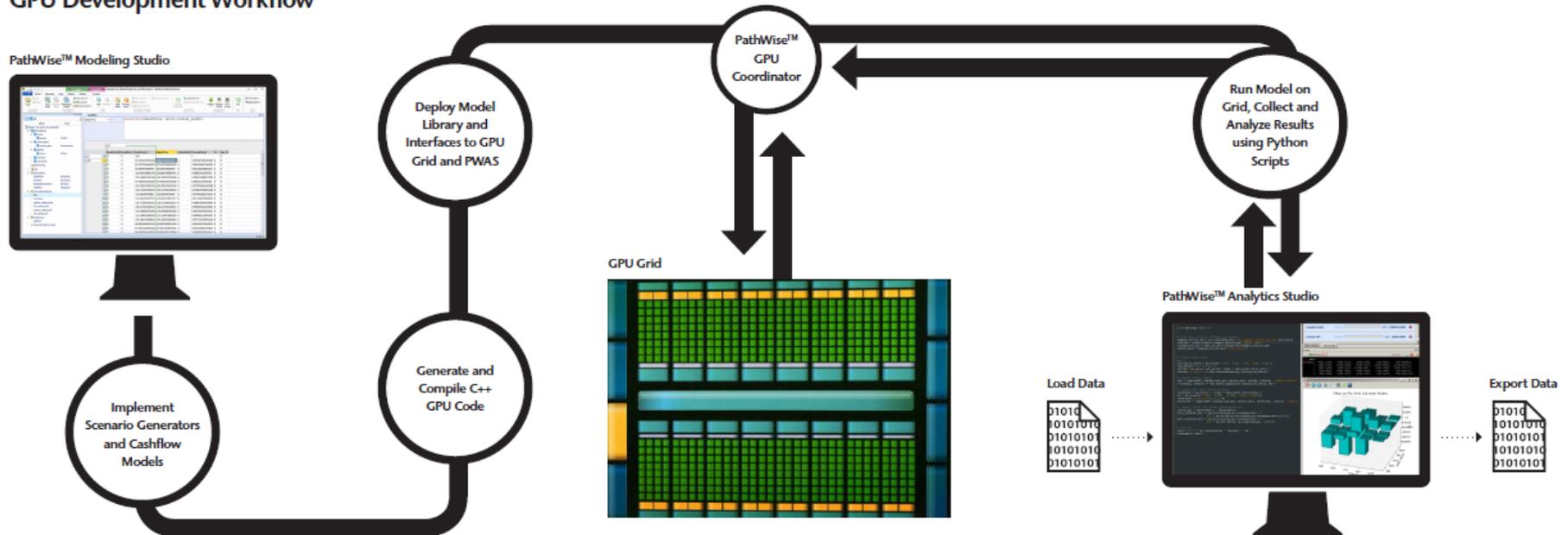


- PathWise won IDC's HPC Innovation Excellence Award in June 2012
“The new award winners and project leaders announced at ISC'12 are as follows (contact IDC for additional details about the projects):
- **GE Global Research (U.S.).**
- **Department of Defense High Performance Computing Modernization Program (U.S.).**
- **Mary Bird Perkins Cancer Center and Louisiana State University (U.S.).**
- **BGI Shenzhen (China).**
- **Aon Benfield Securities, Inc. (Canada).** Aon has developed the PathWise platform, which uses GPU-based high performance computing to enable quantitative analysts to quickly and easily express financial application kernels such as Monte Carlo simulations using domain-specific interfaces. The computational capabilities offered by the GPU-driven HPC enabled quantitative analysts to accelerate financial computations from days to minutes, with 50-100 times throughput over conventional techniques. The PathWise platform from Aon Benfield achieved an average 90% cost savings both in terms of HPC infrastructure costs and time-to-market, translating to several millions of dollars in savings. Project leader: Peter Phillips, Aamir Mohammad”

PathWise Modeling Studio

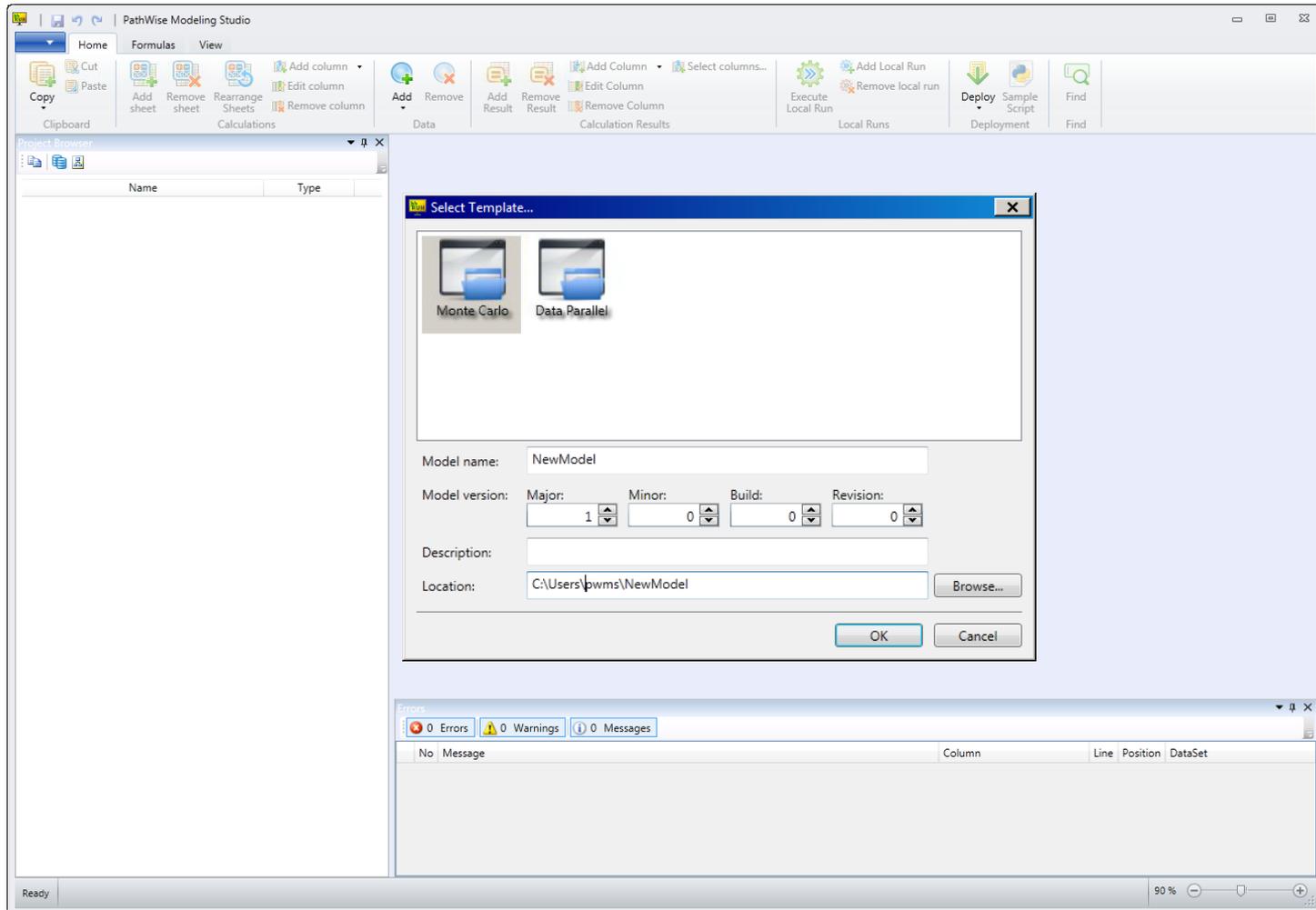
- End-user tools for High Productivity Computing

GPU Development Workflow



PathWise Modeling Studio

- Create a new model



PathWise Modeling Studio

- Define input data structures (customized NumPy data structures)

The screenshot displays the PathWise Modeling Studio interface. The main window shows a project browser on the left with a tree view containing 'Model - GMMB', 'Data sources', 'model_data', 'inforce', 'lapse_tables', 'mortality_tables', 'custom_scenarios', 'market_data', 'shocks', 'Shocks', 'RNG Configuration', 'Model Libraries', 'Calculations', 'Calculation Results', 'Greeks', 'Local Runs', and 'User-defined functions'. The central workspace shows a table with columns 'mortality_table_id', 'age', and 'Qt'. The 'Qt' column is highlighted in yellow. The 'Add/Edit Data Set' dialog box is open, showing the following configuration:

- Name: mortality_tables
- Type: Assumptions
- Alias: (empty)
- Array Size: Fixed (selected)
- Update formulas:
- Data Source: Name: model_data, Path: C:\Documents\pwms\GMMB\data\model_data.npz
- Schema options: Shape: 2, Precision: Float (selected)
- Array type: Structure (selected)
- Type: (empty)
- Table of variables:

Name	Type	Shape	Shock
mortality_table_id	int	None	None
age	int	116	None
Qt	real	116	Auto

The DType is shown as: `[(mortality_table_id, '<i4', ('age', '<i4', (116L)), ('Qt', '<f4', (116L)))]`

PathWise Modeling Studio

- Setup Random Number Generator options

The screenshot shows the PathWise Modeling Studio interface with the 'RNG Configuration' dialog box open. The dialog is titled 'RNG Configuration' and shows settings for a Monte Carlo simulation. The 'Uniform Generator' is set to 'XORWOW Pseudo-random Generator'. The 'Distribution Algorithm' is set to 'Mersenne Twister (607) Pseudo-random Generator'. The 'Correlation Matrix' is set to 'Linear Correlation' and the matrix is previewed below.

Configuration parameters shown in the dialog:

- seed: =inforce.rng_seed
- Num paths: =inforce.num_paths
- Skip rows: =0
- Current path index: < 0 >
- Uniform Generator: XORWOW Pseudo-random Generator
- Seed: =inforce.rng_seed
- Distribution Algorithm: Mersenne Twister (607) Pseudo-random Generator
- Non-Central Chi Square: Sobol (32) Quasi-random Generator, Sobol (64) Quasi-random Generator, Minimal Random Generator, CURAND MTGP11213 Generator, CURAND MRGP32K3A Generator
- Model library RNG combinations: Default
- Correlation: Linear Correlation
- Correlation Matrix: =GBM_HW2F_correlation.cor_matrix

Correlation Matrix Preview:

		Equity_ESG				Interest_Rate_ESG		
		corr_whitenoise[0] (Normal)	corr_whitenoise[1] (Normal)	corr_whitenoise[2] (Normal)	corr_whitenoise[3] (Normal)	corr_whitenoise[4] (Normal)	corr_vn[0] (Normal)	corr_vn[1] (Normal)
EquityESG	corr_whitenoise[0] (Normal)	1	0.822	0.681	0.971	0.733	0	0
	corr_whitenoise[1] (Normal)	0.822	1	0.686	0.806	0.66	0	0
	corr_whitenoise[2] (Normal)	0.681	0.686	1	0.651	0.904	0	0
	corr_whitenoise[3] (Normal)	0.971	0.806	0.651	1	0.68	0	0
	corr_whitenoise[4] (Normal)	0.733	0.66	0.904	0.68	1	0	0
InterestRate_ESG	corr_vn[0] (Normal)	0	0	0	0	0	1	0
	corr_vn[1] (Normal)	0	0	0	0	0	0	1

PathWise Modeling Studio

- Import and configure Model Libraries (e.g. pre-built Economic Scenario Generators)

The screenshot displays the PathWise Modeling Studio interface. The main window shows a data table with columns for time (dt_in_years), correlation (corr_whitenoise), and log returns. A dropdown menu for 'RNG Configuration' is open, listing options: chisquare, ncchisquare, normal, poisson, and uniform. The 'normal' option is selected. The data table shows values for dt_in_years (0 to 25), corr_whitenoise[0] to [4], and log_returns[0] to [3].

dt_in_years	corr_whitenoise[0]	corr_whitenoise[1]	corr_whitenoise[2]	corr_whitenoise[3]	corr_whitenoise[4]	log_returns[0]	log_returns[1]	log_returns[2]	log_returns[3]
0	0.06301	0	0	0	0	0.0000%	0.0000%	0.0000%	0.0000%
1	0.08219	-0.1548973	-0.8779107	0.3820293	-0.4845392	-0.8619%	-1.3759%	-4.4918%	1.8338%
2	0.08219	0.04265733	0.423768	0.2174918	-0.4224617	0.1559%	4.4716%	2.3411%	1.1584%
3	0.08219	-1.006912	0.4300877	-1.285436	0.2424345	-5.8794%	-12.7533%	2.3601%	-7.4764%
4	0.08219	-1.465682	0.3621927	-1.367405	-0.4851904	-8.5016%	-1.7838%	1.9791%	-7.9381%
5	0.08219	-0.09411632	-0.7684383	-1.156138	0.08611837	-0.6681%	-4.5345%	-6.7575%	0.3653%
6	0.08219	-0.7466307	0.1432575	0.1404346	-0.7820051	-4.4784%	0.6241%	0.6079%	-4.6812%
7	0.08219	-0.4416853	1.038652	0.3633792	-0.3015822	-2.6899%	5.7981%	1.9262%	-1.8866%
8	0.08219	-1.062231	-0.06021303	-0.7155739	-0.7718784	-6.1989%	-0.4535%	-4.2113%	-4.5341%
9	0.08219	-0.8860358	0.3283949	0.493393	-0.672089	-5.2046%	1.7588%	2.7048%	-3.9778%
10	0.08219	1.374672	1.129136	0.8991433	1.117549	7.7378%	6.3300%	5.0112%	5.5231%
11	0.08219	0.1007481	0.6669464	0.3477014	0.5162982	0.4677%	3.7142%	1.8837%	2.8504%
12	0.08219	1.954194	1.634924	0.8792632	1.634747	11.1139%	9.2833%	4.9505%	9.2823%
13	0.08219	0.11927	-0.7092768	-0.5051804	0.02469391	0.5843%	-4.1665%	-2.9962%	0.0420%
14	0.08219	0.3938667	0.4305502	0.2574111	0.8597993	2.2269%	2.4372%	1.4445%	4.8985%
15	0.08219	-0.7380639	-1.767081	-0.6726584	-0.7953085	-4.2573%	-10.1575%	-3.8823%	-4.5856%
16	0.08219	-0.4155868	-1.023391	-0.3492023	-0.382019	-2.4471%	-5.9322%	-2.0665%	-2.2547%
17	0.08219	0.7107771	0.03922832	0.3088462	0.6509885	4.0496%	0.1990%	1.7450%	3.7068%
18	0.08219	0.1147579	0.2721997	0.5815271	0.3309397	0.6790%	1.5817%	3.3554%	1.9185%
19	0.08219	1.885169	2.216563	1.807503	1.778597	10.8843%	12.7845%	10.4390%	10.2733%
20	0.08219	-0.8022885	-1.668396	-0.7146163	-0.7122384	-4.5617%	-9.5278%	-4.0590%	-4.0454%
21	0.08219	1.615635	1.41437	1.582458	1.454906	9.2173%	8.0633%	9.0271%	8.2957%
22	0.08219	0.1332374	-0.1539606	0.1047874	0.1320681	0.7687%	-0.8781%	0.6056%	0.7620%
23	0.08219	2.005604	0.9153425	-0.01529968	2.061847	11.5071%	5.2558%	-0.0804%	11.8296%
24	0.08219	-0.5012379	0.1087353	0.09050599	-1.097984	-2.9234%	0.5741%	0.4696%	-6.3450%
25	0.08219	-0.9138255	-1.407303	-0.3786871	-1.010633	-5.2820%	-8.1115%	-2.2136%	-5.8371%

PathWise Modeling Studio

- Calculate number of time-steps to simulate

The screenshot shows the PathWise Modeling Studio interface. The main window displays a table with columns for simulation parameters. A red circle highlights the 'Time' column, and a red arrow points to the 'Time' label. The table contains the following data:

	prev_projection_date	current_projection_date	next_projection_date	prev_projection_date_jd	current_projection_date_jd	projection_time_in_days	projection_time_in_y
0	2012-01-01	2012-01-01	2012-01-24	2455928	2455928	0	0.00000
1	2012-01-01	2012-01-24	2012-02-23	2455928	2455951	23	0.06301
2	2012-01-24	2012-02-23	2012-03-24	2455951	2455981	53	0.14521
3	2012-02-23	2012-03-24	2012-04-23	2455981	2456011	83	0.22740
4	2012-03-24	2012-04-23	2012-05-23	2456011	2456041	113	0.30959
5	2012-04-23	2012-05-23	2012-06-22	2456041	2456071	143	0.39178
6	2012-05-23	2012-06-22	2012-07-22	2456071	2456101	173	0.47397
7	2012-06-22	2012-07-22	2012-08-21	2456101	2456131	203	0.55616
8	2012-07-22	2012-08-21	2012-09-20	2456131	2456161	233	0.63836
9	2012-08-21	2012-09-20	2012-10-20	2456161	2456191	263	0.72055
10	2012-09-20	2012-10-20	2012-11-19	2456191	2456221	293	0.80274
11	2012-10-20	2012-11-19	2012-12-19	2456221	2456251	323	0.88493
12	2012-11-19	2012-12-19	2013-01-18	2456251	2456281	353	0.96712
13	2012-12-19	2013-01-18	2013-02-17	2456281	2456311	383	1.04932
14	2013-01-18	2013-02-17	2013-03-19	2456311	2456341	413	1.13151
15	2013-02-17	2013-03-19	2013-04-18	2456341	2456371	443	1.21370
16	2013-03-19	2013-04-18	2013-05-18	2456371	2456401	473	1.29589
17	2013-04-18	2013-05-18	2013-06-17	2456401	2456431	503	1.37808
18	2013-05-18	2013-06-17	2013-07-17	2456431	2456461	533	1.46027
19	2013-06-17	2013-07-17	2013-08-16	2456461	2456491	563	1.54247
20	2013-07-17	2013-08-16	2013-09-15	2456491	2456521	593	1.62466
21	2013-08-16	2013-09-15	2013-10-15	2456521	2456551	623	1.70685
22	2013-09-15	2013-10-15	2013-11-14	2456551	2456581	653	1.78904
23	2013-10-15	2013-11-14	2013-12-14	2456581	2456611	683	1.87123
24	2013-11-14	2013-12-14	2014-01-13	2456611	2456641	713	1.95343
25	2013-12-14	2014-01-13	2014-02-12	2456641	2456671	743	2.03562

PathWise Modeling Studio

- Define simulation columns and formulas

The screenshot displays the PathWise Modeling Studio interface. The main window shows a spreadsheet with columns for simulation parameters and results. A formula editor is open, showing a conditional formula for 'guaranteed_value'.

Formula Editor:

```

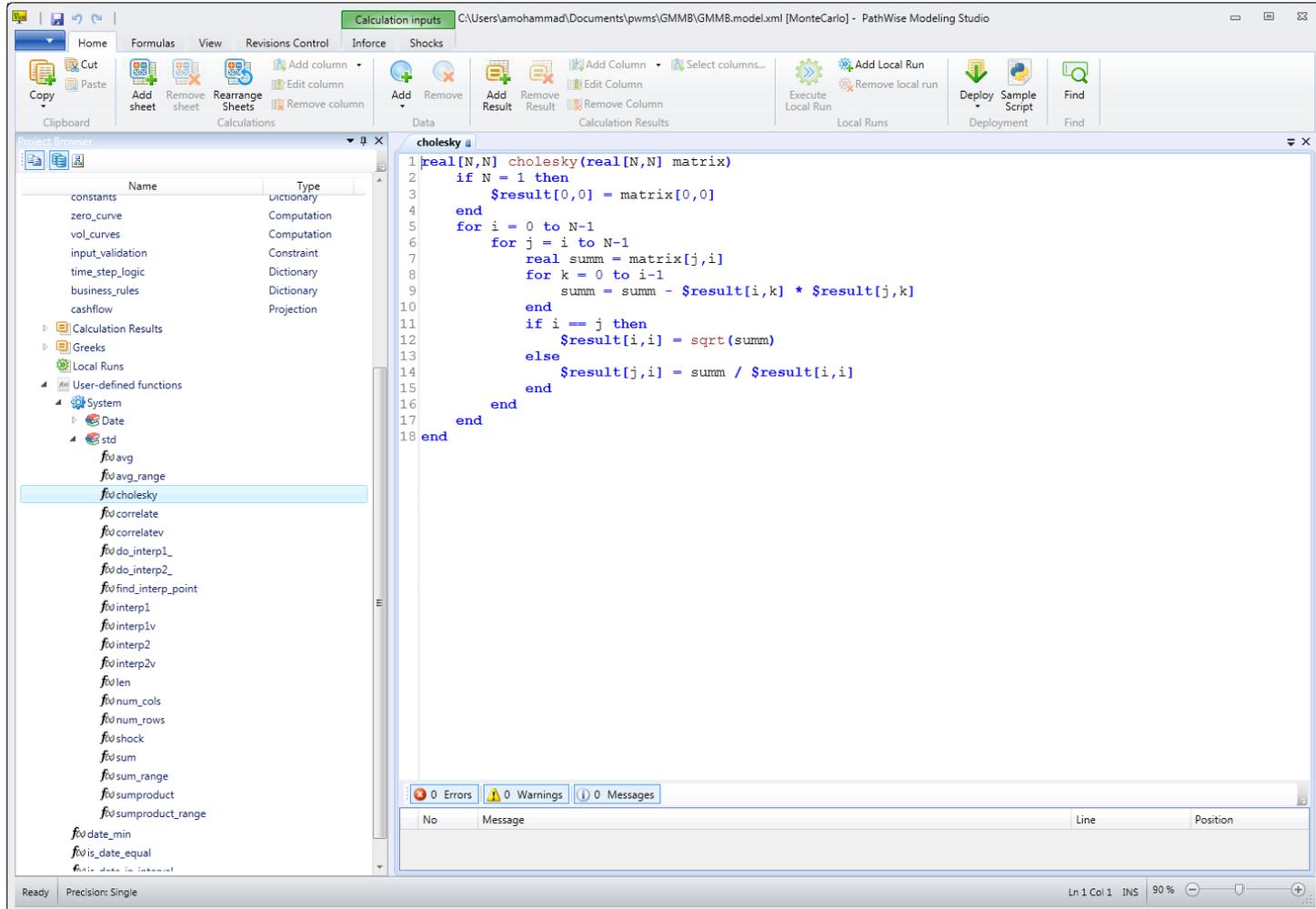
if is_ratchet_step then
  gv = max(total_account_value, prev(guaranteed_value))
else
  gv = prev(guaranteed_value)
endif
return gv
  
```

Spreadsheet Data (Columns):

	_return[2]	account_value[2]	log_r		_return[4]	account_value[4]	total_account_value	guaranteed_value	itm_pct	base_la
1	0.20%	\$00.00	0.00		000%	\$00.00	\$100,000.00	\$100,000.00	0.00%	0.00%
122	1.92%	\$00.00	1.83		517%	\$00.00	\$99,141.83	\$100,000.00	0.70%	1.00%
	2.41%	\$00.00	1.15		511%	\$00.00	\$99,296.53	\$100,000.00	0.70%	1.00%
	3.50%	\$00.00	-7.47		284%	\$00.00	\$93,626.80	\$100,000.00	6.37%	1.00%
	4.79%	\$00.00	-7.95		880%	\$00.00	\$85,995.97	\$100,000.00	14.00%	1.00%
	5.75%	\$00.00	0.36		551%	\$00.00	\$85,423.35	\$100,000.00	14.58%	1.00%
	6.38%	\$00.00	-4.68		208%	\$00.00	\$81,682.17	\$100,000.00	18.32%	1.00%
	7.26%	\$00.00	-1.88		328%	\$00.00	\$79,514.26	\$100,000.00	20.49%	1.00%
	8.11%	\$00.00	-4.57%	\$00.00	-5.04%	\$00.00	\$74,734.88	\$100,000.00	25.27%	1.00%
	9.05%	\$00.00	-3.97%	\$00.00	0.089%	\$00.00	\$70,944.74	\$100,000.00	29.06%	1.00%
	10.11%	\$00.00	5.52%	\$00.00	6.26%	\$00.00	\$76,652.30	\$100,000.00	23.35%	1.00%
	11.34%	\$00.00	2.85%	\$00.00	-2.657%	\$00.00	\$77,011.66	\$100,000.00	22.99%	1.00%
	12.50%	\$00.00	9.28%	\$00.00	5.56%	\$00.00	\$86,064.43	\$100,000.00	13.94%	1.00%
	13.96%	\$00.00	0.04%	\$00.00	-0.154%	\$00.00	\$86,568.73	\$100,000.00	13.43%	2.00%
	14.44%	\$00.00	4.89%	\$00.00	2.35%	\$00.00	\$88,518.16	\$100,000.00	11.48%	2.00%
	15.82%	\$00.00	-4.58%	\$00.00	-3.459%	\$00.00	\$84,828.74	\$100,000.00	15.17%	2.00%
	16.66%	\$00.00	-2.25%	\$00.00	-0.382%	\$00.00	\$82,778.07	\$100,000.00	17.22%	2.00%
	17.45%	\$00.00	3.70%	\$00.00	0.950%	\$00.00	\$86,199.04	\$100,000.00	13.80%	2.00%
	18.55%	\$00.00	1.91%	\$00.00	1.457%	\$00.00	\$86,786.30	\$100,000.00	13.21%	2.00%
	19.439%	\$00.00	10.27%	\$00.00	13.510%	\$00.00	\$96,765.66	\$100,000.00	3.23%	2.00%
	20.59%	\$00.00	-4.04%	\$00.00	-7.236%	\$00.00	\$92,450.63	\$100,000.00	7.55%	2.00%
	21.27%	\$00.00	8.29%	\$00.00	7.582%	\$00.00	\$101,377.10	\$100,000.00	-1.38%	2.00%
	22.06%	\$00.00	0.76%	\$00.00	0.020%	\$00.00	\$102,159.40	\$100,000.00	-2.16%	2.00%
	23.80%	\$00.00	11.830%	\$00.00	3.03%	\$00.00	\$114,618.10	\$100,000.00	-14.62%	2.00%
	24.70%	\$00.00	-6.345%	\$00.00	3.837%	\$00.00	\$111,315.90	\$100,000.00	-11.32%	2.00%
	25.14%	\$00.00	-5.837%	\$00.00	-2.908%	\$00.00	\$105,588.80	\$100,000.00	-5.59%	3.00%

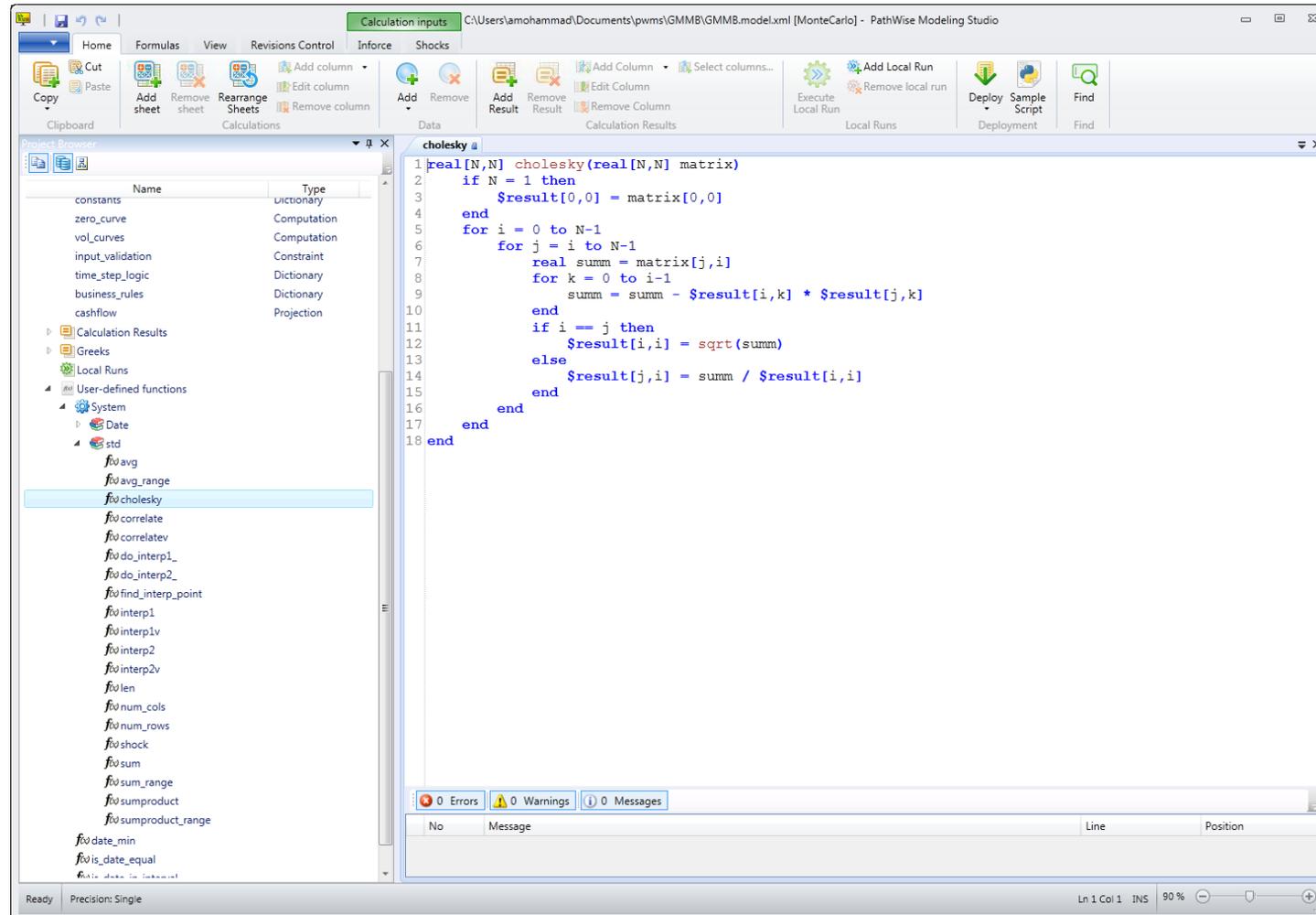
PathWise Modeling Studio

- Encapsulate re-usable logic in UDFs and UDF libraries



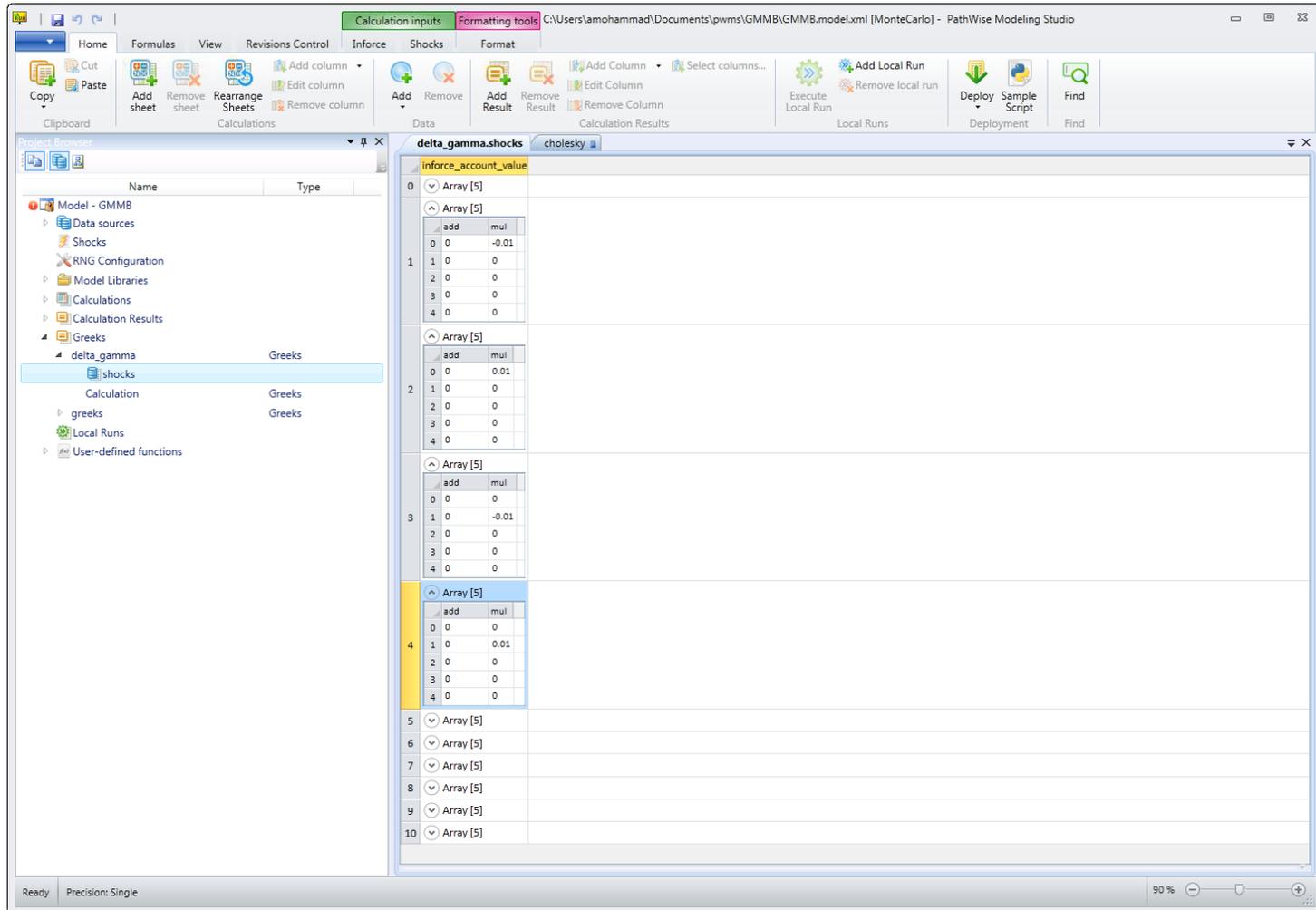
PathWise Modeling Studio

- Encapsulate re-usable logic in UDFs and UDF libraries



PathWise Modeling Studio

- Define model outputs (e.g. Greeks)



PathWise Modeling Studio

- Define model outputs (e.g. Greeks)

The screenshot displays the PathWise Modeling Studio interface. The top menu bar includes Home, Formulas, View, Revisions Control, Inforce, and Shocks. The main workspace is divided into a Project Browser on the left and a central editor area. The Project Browser shows a tree structure for 'Model - GMMB', including 'Data sources', 'Shocks', 'RNG Configuration', 'Model Libraries', 'Calculations', 'Calculation Results', 'Greeks', 'Local Runs', and 'User-defined functions'. Under 'Greeks', 'delta_gamma' is expanded to show 'shocks' and 'Calculation'. The central editor shows the definition of the 'delta' calculation, with a formula field containing the following code:

```

idx_dn = $index*2 + 1
idx_up = $index*2 + 2
fmv_up = fmv[idx_up].avg_fmv
fmv_dn = fmv[idx_dn].avg_fmv
av_up = shock( inforce.account_value[$index], shocks[idx_up].inforce_account_value[$index] )
av_dn = shock( inforce.account_value[$index], shocks[idx_dn].inforce_account_value[$index] )
return ( fmv_up - fmv_dn ) / ( av_up - av_dn )

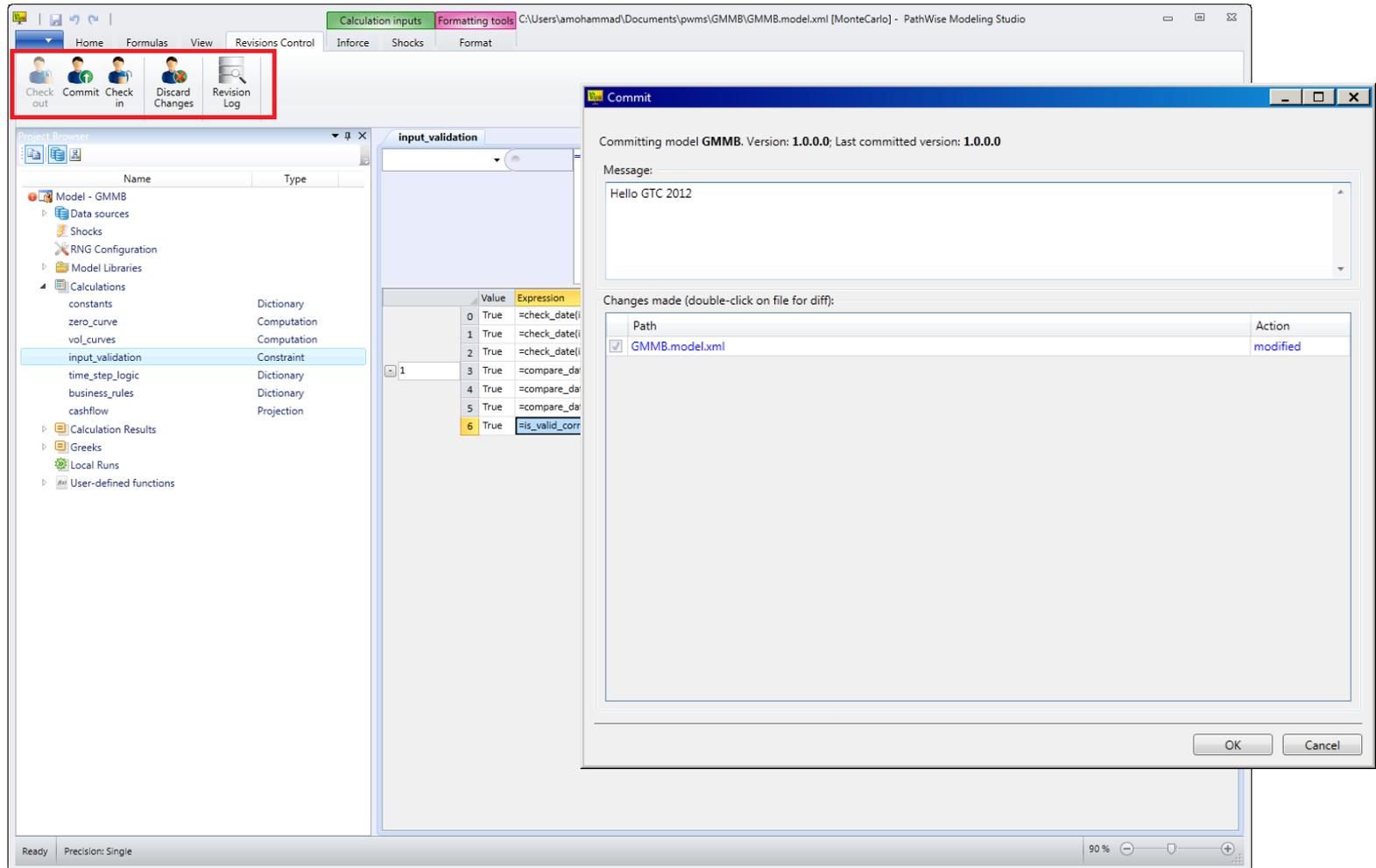
```

Below the editor, a table displays the calculated values for various Greek outputs. The table has three columns: 'name', 'formula', and 'calculated value'. The 'name' column includes 'base_fmv', 'delta[0]', 'dollar_delta[0]', 'gamma[0]', and 'dollar_gamma[0]'. The 'formula' column contains the corresponding formulas for each output. The 'calculated value' column shows the results: -72.0284 for base_fmv, 0.00425906 for delta[0], 425.906 for dollar_delta[0], -2.408791E-08 for gamma[0], and -240.879 for dollar_gamma[0].

name	formula	calculated value
base_fmv	=fmv[0].avg_fmv	-72.0284
delta[0]	idx_dn = \$index*2 + 1 idx_up = \$index*2 + 2 fmv_up = fmv[idx_up].avg_fmv fmv_dn = fmv[idx_dn].avg_fmv av_up = shock(inforce.account_value[\$index], shocks[idx_up].inforce_account_value[\$index]) av_dn = shock(inforce.account_value[\$index], shocks[idx_dn].inforce_account_value[\$index]) return (fmv_up - fmv_dn) / (av_up - av_dn)	0.00425906
dollar_delta[0]	idx_dn = \$index*2 + 1 idx_up = \$index*2 + 2 fmv_up = fmv[idx_up].avg_fmv fmv_dn = fmv[idx_dn].avg_fmv av_up = shock(inforce.account_value[\$index], shocks[idx_up].inforce_account_value[\$index]) av_dn = shock(inforce.account_value[\$index], shocks[idx_dn].inforce_account_value[\$index]) av_base = inforce.account_value[\$index] return (fmv_up - fmv_dn) / (av_up - av_dn) * av_base	425.906
gamma[0]	idx_dn = \$index*2 + 1 idx_up = \$index*2 + 2 fmv_up = fmv[idx_up].avg_fmv fmv_dn = fmv[idx_dn].avg_fmv fmv_base = fmv[0].avg_fmv av_up = shock(inforce.account_value[\$index], shocks[idx_up].inforce_account_value[\$index]) av_dn = shock(inforce.account_value[\$index], shocks[idx_dn].inforce_account_value[\$index]) return (fmv_up - 2*fmv_base + fmv_dn) / ((av_up - av_dn)^2)	-2.408791E-08
dollar_gamma[0]	idx_dn = \$index*2 + 1 idx_up = \$index*2 + 2 fmv_up = fmv[idx_up].avg_fmv fmv_dn = fmv[idx_dn].avg_fmv fmv_base = fmv[0].avg_fmv av_up = shock(inforce.account_value[\$index], shocks[idx_up].inforce_account_value[\$index]) av_dn = shock(inforce.account_value[\$index], shocks[idx_dn].inforce_account_value[\$index]) av_base = inforce.account_value[\$index] return (fmv_up - 2*fmv_base + fmv_dn) / ((av_up - av_dn)^2) * av_base^2	-240.879

PathWise Modeling Studio

- Commit model to SVN source code repo



PathWise Modeling Studio

■ Compile and deploy model to GPUs

The screenshot displays the PathWise Modeling Studio interface. The ribbon at the top includes tabs for 'Home', 'Formulas', 'View', 'Revisions Control', 'Inforce', 'Shocks', and 'Format'. The 'Deploy' button, represented by a green downward arrow, is circled in red. A 'Deploying...' dialog box is open, showing a list of messages:

- Info: C++ Project generation started for model 'GMMB'
- Warning: ..Column: 'hw2_parameters_hw2_drift' not used - removed
- Info: Generating c++ gpud interface
- Info: Generating python interface
- Info: Uploading model C++ sources to Linux Build Computer
- Info: Compiling model C++ sources
- Info: Uploading compiled model to /C:/Users/amohammad/AppData/Local/Temp/5/pwms_temp/GMMB_v1_0_0/work_temp
- Info: Calculating C:\Users\amohammad\AppData\Local\Temp\5\pwms_temp\GMMB_v1_0_0\work_temp\GMMB_v1_0_0.model:
- Info: Uploading Python interfaces for PwaStudio
- Info: Uploading py to \\AMZXA01.asgpn.net\DataProd#asg_600\PwasGridDeploymentRegistry\ (overwriting existing files)
- Info: Restarting grid. Please wait...
- Info: Requesting disconnect of 2 worker(s)
- Info: Waiting up to 60 seconds for workers to come back online
- Info: Grid restarted
- Info: Deployment completed

The dialog box also features a 'Debug Mode' checkbox, 'Edit Deployment Configuration', 'Copy to Clipboard', and 'Close' buttons. At the bottom of the dialog, a green circle indicates the status is 'Ready'.

The background spreadsheet shows a table with columns for 'fwd_rate' and various numerical values. The 'fwd_rate' column contains values ranging from 0.00% to 1.69%. The numerical values in the adjacent columns range from \$0.00 to \$111,315.90.

PathWise Modeling Studio

- Add GPU grid workers from the **Cloud**

The screenshot displays the PathWise Grid Management Console interface. At the top, there are tabs for 'session', 'workers (2/6)', 'models (4)', 'amazon workers', and 'settings'. Below the tabs, there is a 'Select All' checkbox and two buttons: 'Start new instances' and 'Terminate selected instances'. A table lists worker details:

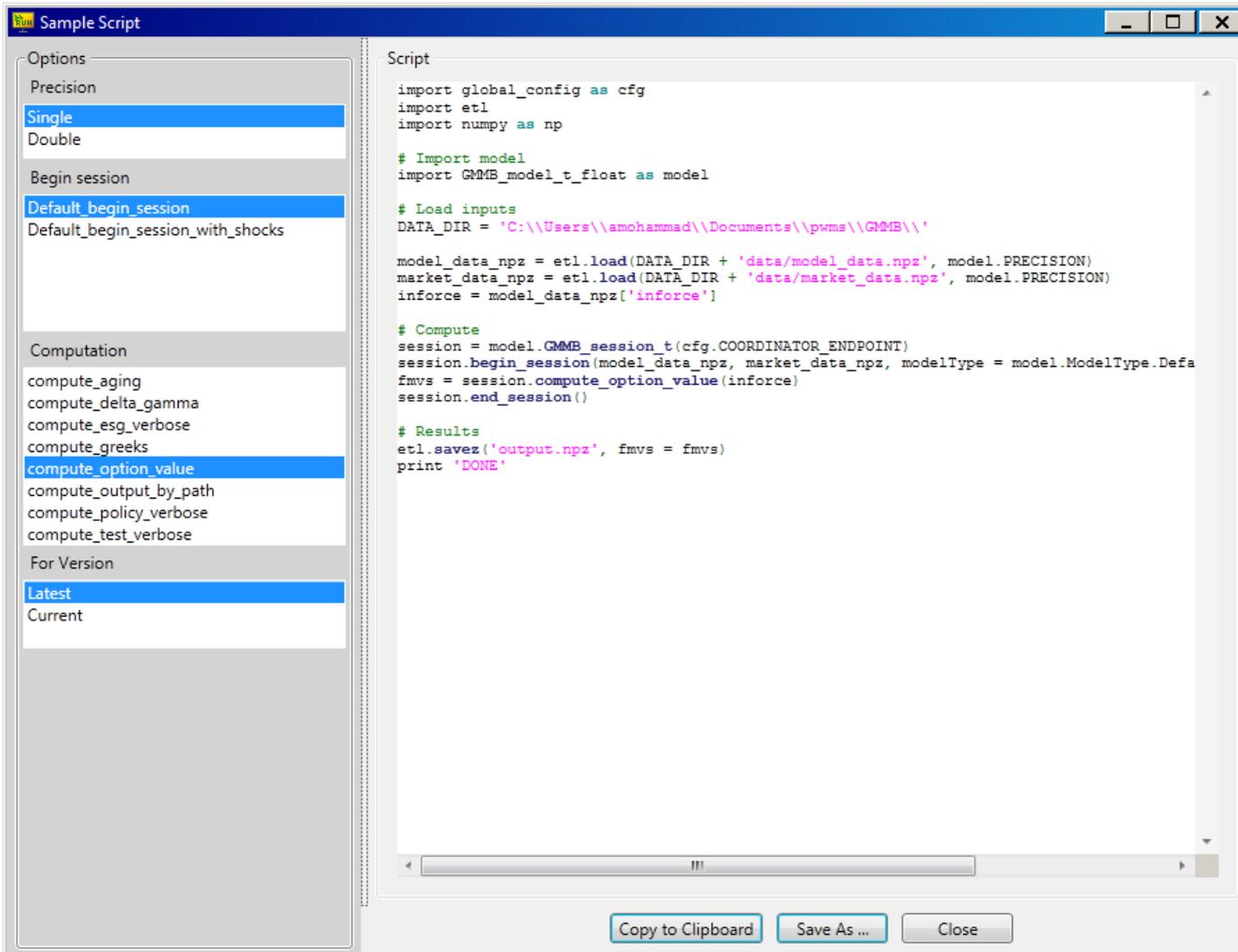
ip address	state	launch time	time running	termination policy	instance type	max price	started by
10.16.0.205	Online	5/10/2012 10:55:40 AM	09h 42m 46s	Never	Spot	\$1.500	ASGPW\svcpathwise

A 'Start Instances' dialog box is open, showing options for 'Termination policy' (Never or After 1 idle hour), 'On-Demand instances', and 'Spot instances'. The 'Spot instances' section shows a 'Current price: \$0.665', a 'Max Price' input field with '0.665', and a 'Spot instances count' input field with '1'. A 'Start Spot instances' button is at the bottom of the dialog.

At the bottom of the console, it shows 'Current spot price: \$0.665' and a status bar with '10.17.163.175:6147' and 'last update : 20:38:32'.

PathWise Modeling Studio

- Generate sample Python script



PathWise Analytics Studio

- Run Python scripts from **PathWise Analytics Studio** (customized Python IDE)

The screenshot displays the PathWise Analytics Studio 2.0 interface. On the left, a Python script editor shows code for loading data, creating shock arrays, and computing P&L attribution. The right side features a task manager with progress bars for 'Compute Greeks' and 'Compute FMV'. Below the task manager is a console window and a 3D bar chart titled 'P&L Attribution, 2nd Order Greeks'. The chart shows P&L values for various market movements (MM, DEX, TSX, SPX, EAFE) and index shocks (Index Shock, Index Shock).

```
1 ## Copyright (C) Aon Benfield Securities 2012, All Rights Reserved
2 from settings import *
3
4 # Load data (inforce, assumptions, market)
5 mapped_inforce_npz = etl.load(DATA_DIR+'signed_mapped_inforce-20120222.npz',precision)
6 inforce = scale_inforce( mapped_inforce_npz, 20000, 41632 )
7 assumptions_npz = demo.get_best_assumptions(mapped_inforce_npz)
8 market_data = mapped_inforce_npz['market_data']
9
10 # Create shock array
11 n = 5
12 multiplier_shock = np.array([ 0.01, 0.01, 0.01, 0.01, 0.01 ])
13 num_shocks = 2 * n * n + 1
14 shocks = np.zeros( num_shocks, dtype=demo_model.shock_data_t )
15 shocks['av']['mul'] = shk.createShockArray( multiplier_shock )
16
17 # Compute Delta & Gamma
18 fmv = computeFMV( assumptions_npz, market_data, shocks, inforce, 'Compute Greeks' )
19 (jacobian, hessian) = shk.taylor_expansion( multiplier_shock, fmv )
20
21 # Compute FMV under market movement
22 newshocks = np.zeros( 2, dtype = demo_model.shock_data_t )
23 dx = np.array([[ 0.007, 0.005, -0.005, 0.01, 0.008 ]])
24 newshocks['av']['mul'][1,:] = dx
25 marketfmv = computeFMV( assumptions_npz, market_data, newshocks, inforce, 'Compute FMV' )
26
27 # Compare actual PnL versus approximated PnL
28 actual_pnl = marketfmv[1] - marketfmv[0]
29 full_2ndorder_pnl = (np.dot(jacobian,np.transpose(dx)) + \
30 0.5 * np.dot(dx,np.dot(hessian,np.transpose(dx)))) [0][0]
31 part_2ndorder_pnl = (np.dot(jacobian,np.transpose(dx)) + \
32 0.5 * np.dot( dx*dx, np.diag(hessian) )) [0][0]
33
34 # Plot PnL
35 data = 0.5 * ( np.transpose(dx) * hessian ) * dx
36 createplot( data )
```

Task manager:

- Compute Greeks: 00:04:03, 100%, 1020000/1020000
- Compute FMV: 00:01:15, 100%, 40000/40000

Figure 1: P&L Attribution, 2nd Order Greeks

3D Bar Chart Data (Approximate):

Category	P&L Value
MM	~10000
DEX	~10000
TSX	~10000
SPX	~10000
EAFE	~10000
Index Shock	~10000

Console: Figure 1

Object inspector: Task manager

Permissions: RW | End-of-lines: CRLF | Encoding: ASCII | Line: 14 | Column: 37

PathWise Analytics Studio

- Run Python scripts from **PathWise Analytics Studio** (customized Python IDE)

The screenshot displays the PathWise Analytics Studio 2.0 interface. The main window is titled "PathWise Analytics Studio 2.0" and contains a Python script editor on the left, a task manager on the right, and a console window at the bottom right. The script editor shows a Python script with the following code:

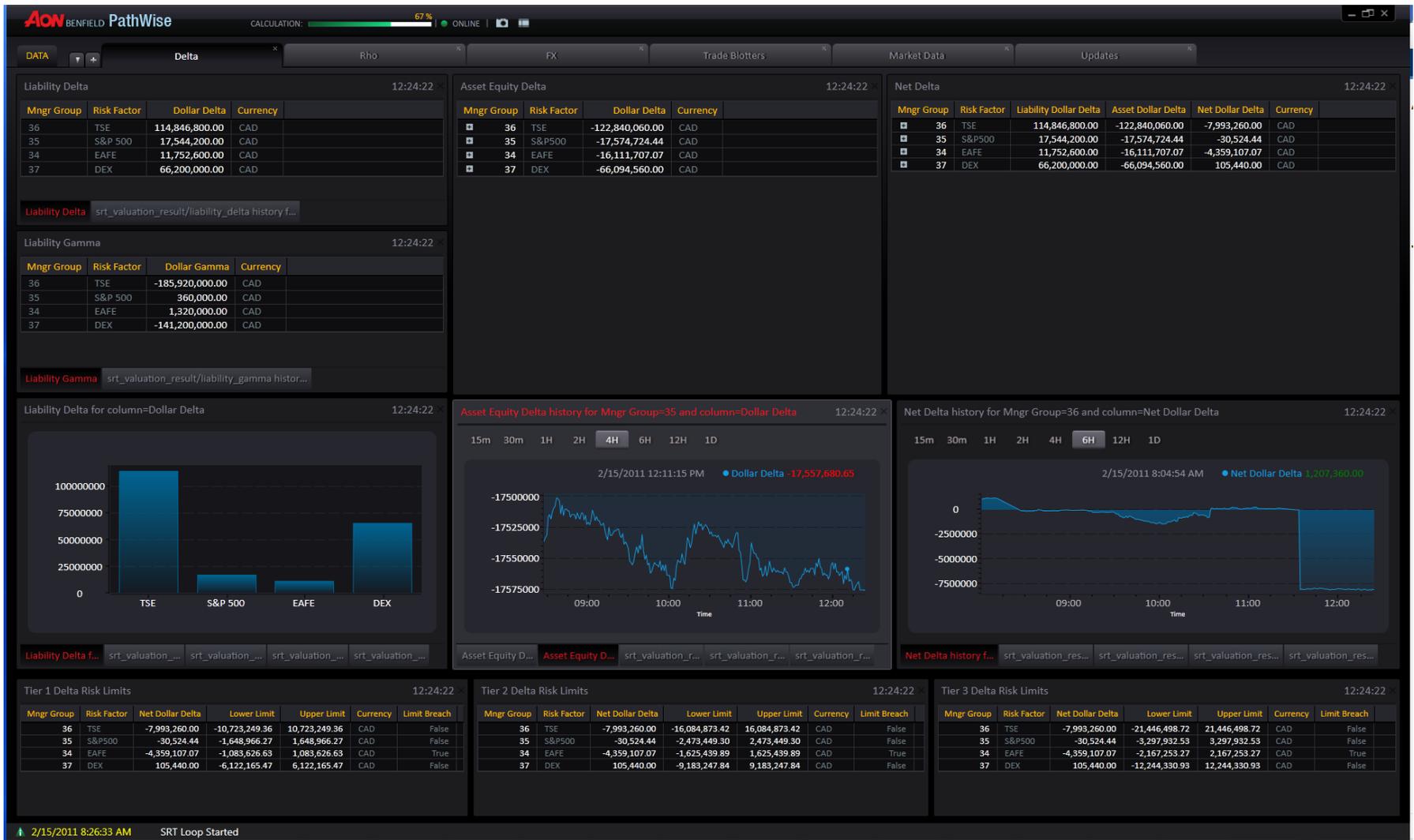
```
3 import numpy as np
4 import utils
5
6 # import model
7 import GMM
8 import GMM
9
10 NUM_INDICES
11
12 # Load input
13 DATA_DIR =
14
15 model_data
16 market_data
17 shocks_npz
18 inforce = n
19
20 # modify in
21 inforce['nu
22
23 inforce['ts
24 inforce['ts
25
26 inforce['ac
27 inforce['gl
28 inforce['ra
29
30 # create sh
31 num_shocks
32 min_av_shoc
33 max_av_shoc
34 shock_grid
35
36 shocks_npz
```

The task manager window shows two tasks: "GMMB_session_t.compute_delta_gamma" with a progress bar at 100% and a status of "1/1". The console window shows the output of the script, including the number of workers and the output of a print statement:

```
NUM_WORKERS: 10
>>> print 'Hello GTC 2012'
Hello GTC 2012
>>>
```

The main window also displays two plots. The top plot, titled "Delta Sensitivity", shows Delta on the y-axis (ranging from 0.00 to 0.30) versus Account Value Shock (%) on the x-axis (ranging from -60 to 60). The bottom plot, titled "Gamma Sensitivity", shows Gamma on the y-axis (ranging from -0.0000014 to 0.0000000) versus Account Value Shock (%) on the x-axis (ranging from -60 to 60).

PathWise Seriatim Real-Time Risk System



Thank You